Pre-Mid(Task#1)

Submitted to: Sir Aqib

Submitted by: Hifza Khalid

Roll#:SU92_BSSEM_F22_202

Subject: Advance Computer Programming

Section: BSSE-4D

Date: Jan8,2024.

Code:

```
import java.util.Scanner;

class MatrixOperations {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Create two matrices
        int[][] matrix1 = {
                {1, 2, 3},
                {4, 5, 6},
                {7, 8, 9}
        };

        int[][] matrix2 = {
                {9, 8, 7},
                {6, 5, 4},
                {3, 2, 1}
        };

        // Display menu
        while (true) {
            System.out.println("    ***Matrix Operations Menu***");
            System.out.println("           1. Addition");
            System.out.println("           2. Subtraction");
            System.out.println("           3. Transposition");
            System.out.println("           4. Scalar Multiplication");
```

```
            System.out.println("            5. Exit");
            System.out.print("Enter your choice (1-5): ");

            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    // Check if matrices have the same order
                    if (hasSameOrder(matrix1, matrix2)) {
                        // Perform matrix addition
                        int[][] sumMatrix = addMatrices(matrix1, matrix2);
                        System.out.println("\nSum of Matrices:");
                        printMatrix(sumMatrix);
                    } else {
                        System.out.println("Matrices must have the same order for
addition.");
                    }
                    break;
                case 2:
                    // Check if matrices have the same order
                    if (hasSameOrder(matrix1, matrix2)) {
                        // Perform matrix subtraction
                        int[][] subtractMatrix = subtractMatrices(matrix1,
matrix2);
                        System.out.println("\nSubtraction of Matrices:");
                        printMatrix(subtractMatrix);
                    } else {
                        System.out.println("Matrices must have the same order for
subtraction.");
                    }
                    break;
                case 3:
                    // Perform matrix transposition
                    int[][] transposeMatrix1 = transposeMatrix(matrix1);
                    int[][] transposeMatrix2 = transposeMatrix(matrix2);
                    System.out.println("\nTranspose of Matrix 1:");
                    printMatrix(transposeMatrix1);
                    System.out.println("\nTranspose of Matrix 2:");
                    printMatrix(transposeMatrix2);
                    break;
                case 4:
                    // Perform scalar multiplication
                    System.out.print("Enter scalar value: ");
                    int scalar = scanner.nextInt();
                    int[][] scalarMultiplicationMatrix = scalarMultiply(matrix1,
scalar);
                    System.out.println("\nScalar Multiplication of Matrix 1 by " +
scalar + ":");
                    printMatrix(scalarMultiplicationMatrix);
                    break;
                case 5:
                    // Exit the loop
                    scanner.close();
                    System.exit(0);
                default:
                    System.out.println("Invalid choice. Please enter a number
between 1 and 5.");
            }
        }
    }

    // Helper method to check if two matrices have the same order
    private static boolean hasSameOrder(int[][] matrix1, int[][] matrix2) {
        return matrix1.length == matrix2.length && matrix1[0].length ==
matrix2[0].length;
    }

    // Helper method to add two matrices
```

```
    private static int[][] addMatrices(int[][] matrix1, int[][] matrix2) {
        int rows = matrix1.length;
        int cols = matrix1[0].length;

        int[][] resultMatrix = new int[rows][cols];

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                resultMatrix[i][j] = matrix1[i][j] + matrix2[i][j];
            }
        }

        return resultMatrix;
    }

    // Helper method to subtract two matrices
    private static int[][] subtractMatrices(int[][] matrix1, int[][] matrix2) {
        int rows = matrix1.length;
        int cols = matrix1[0].length;

        int[][] resultMatrix = new int[rows][cols];

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                resultMatrix[i][j] = matrix1[i][j] - matrix2[i][j];
            }
        }

        return resultMatrix;
    }

    // Helper method to transpose a matrix
    private static int[][] transposeMatrix(int[][] matrix) {
        int rows = matrix.length;
        int cols = matrix[0].length;

        int[][] resultMatrix = new int[cols][rows];

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                resultMatrix[j][i] = matrix[i][j];
            }
        }

        return resultMatrix;
    }

    // Helper method to perform scalar multiplication on a matrix
    private static int[][] scalarMultiply(int[][] matrix, int scalar) {
        int rows = matrix.length;
        int cols = matrix[0].length;

        int[][] resultMatrix = new int[rows][cols];

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                resultMatrix[i][j] = matrix[i][j] * scalar;
            }
        }

        return resultMatrix;
    }

    // Helper method to print a matrix
    private static void printMatrix(int[][] matrix) {
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[i].length; j++) {
                System.out.print(matrix[i][j] + " ");
```

```
            }
            System.out.println();
        }
        System.out.println();
    }
}
```

Output:

- **Main Menu:**

```
   ***Matrix Operations Menu***
           1. Addition
           2. Subtraction
           3. Transposition
           4. Scalar Multiplication
            5. Exit
Enter your choice (1-5):
```

- **Addition Operation:**

```
Enter your choice (1-5): 1

Sum of Matrices:
10 10 10
10 10 10
10 10 10
```

- **Subtraction Operation:**

```
Subtraction of Matrices:
-8 -6 -4
-2 0 2
4 6 8
```

- **Transpose Operation:**

```
Enter your choice (1-5): 3

Transpose of Matrix 1:
1 4 7
2 5 8
3 6 9
```

```
Transpose of Matrix 2:
9 6 3
8 5 2
7 4 1
```

- **Scalar Multiplication by number:**

```
Enter scalar value: 3

Scalar Multiplication of Matrix 1 by 3:
3 6 9
12 15 18
21 24 27
```

- **After Completion:**

```
Enter your choice (1-5): 5

Process finished with exit code 0
```